日本産業技術教育学会誌 第53巻第3号 (2011) 179～187

状態遷移概念を利用した制御プログラミングの学習効果†

Learning Effect of Robot Control Programming with State Transition Concept

井戸坂 幸男* 青 木 浩 幸** 李 元 揆*** 久 野 靖**** 兼 宗 進*****
Yukio IDOSAKA Hiroyuki AOKI WonGyu LEE Yasushi KUNO Susumu KANEMUNE

中学校「技術・家庭」における計測・制御学習では，自律型ロボットをプログラムにより制御する実習が予想され，学習用の教材も市販されている。市販されているロボット教材は手続きの考え方に基づくビジュアルプログラミング環境が多く，フローチャートやタイルにより記述するプログラミングが中心である。本研究では，これらの手続きの考え方とは異なる状態遷移の考え方を取り入れたプログラミング方法の有効性を検証した。使用した制御ソフトウェアである Rtoys は，教育用のプログラミング言語であるスクイーク Etoys をロボット制御に対応させ，状態遷移の考え方を取り入れたプログラミングができるようにしたものである。この Rtoys による制御学習の結果，自律型ロボットの触覚センサが接触するたびに動きを変える課題において，状態を意識しない手続きの考え方に基づくプログラミングに比べ，課題の達成率によい結果が得られた。プログラム作品においても，多くの動作を整理してプログラミングできていることが確認できた。状態遷移の考え方を取り入れたプログラミング方法は，有効な方法であることが分かった。

キーワード：計測・制御，自律型ロボット，スクイーク Etoys，プログラミング

1. はじめに

現代の社会生活においては,身近にある多くの電化製品にマイコン制御が組み込まれている。身近になった制御の仕組みを学習することは,生徒にとっても学習意義を感じる内容であると考えられる。2008 年告示，2012 年度完全実施される新しい中学校学習指導要領 [1] では，技術・家庭科において「プログラムによる計測・制御」の学習（以下，「計測・制御学習」と記す）が必修化されることになった。計測・制御学習の効果については複数の先行研究がある。紅林ら [2] は，制御プログラム学習の経験の有無によって,制御に関わる事故報道の制御対象の認知に差が生じることを報告している。古平ら [3] は，プログラムによる計測・制御学習が，身のまわりの機器に目を向けさせ,制御機器の仕組みの理解につながることを報告している。

計測・制御学習で扱う教材は，様々な種類が考えられるが，車型の自律型ロボットが多く使われている [4)-7)]。自律型ロボットは一般に,ロボットに付属するソフトウェアでプログラムを作り,そのプログラムをロボットに転送する。ロボットは転送されたプログラムをロボットだけで処理し，動作する。これらの教材には，手続きの考え方に基づくビジュアルプログラミング環境が多い。ここでの手続きの考え方とは，命令（指令）を順番に実行することを基本とし,その順番を条件判断や繰り返しの構造によって制御していく方法をいう。

本研究では，計測・制御学習における制御プログラムにおいて,状態遷移の考え方を取り入れたプログラミング方法を提案し,車型の自律型ロボットを使った授業で,その有効性を検証する。

2. 手続きの考え方に基づくプログラムの特徴と問題点

2010 年現在，市販されている計測・制御学習用の教材を中心に,手続きの考え方に基づくプログラムの特徴

(2011 年 3 月 31 日受付，2011 年 8 月 31 日受理)
* 松阪市立飯南中学校（現在 松阪市立飯高東中学校)
** 高麗大学（韓国） 院生
*** 高麗大学（韓国)
**** 筑波大学大学院
***** 大阪電気通信大学
† 2010年12月 本学会第28回東海支部大会(愛教大)にて発表

と問題点について検討した。市販されている車型の自律型ロボットでは，赤外線（光）センサやタッチセンサを使った課題が中心となる。代表的な課題としてはライントレースや迷路脱出がある。そこで，ライントレース課題を題材に代表的なプログラムの記述方法を整理する。

2.1　フローチャートによる記述の例

プログラム例（Beauto Builder R [注1)]）として図 1 に示したものは，左右に取り付けられた 2 つの赤外線センサ（左センサ,右センサ）で黒のラインを挟んで走行するライントレースプログラムである。このプログラムは，矢印を使ってプログラムの流れを指定している。反復構造は，矢印を前の命令に戻す方法で表現する。分岐構造は，菱形のブロックから複数の矢印を出すことで表現している。ライントレースでは，センサによる検知を無限に繰り返させる必要があり，一番下の命令ブロックからの矢印を最初の部分に接続したプログラムとなっている。この方法は，図 2 に示したフローチャートと比較すると分かるように，フローチャートをそのままプログラムとして記述した形となっている。

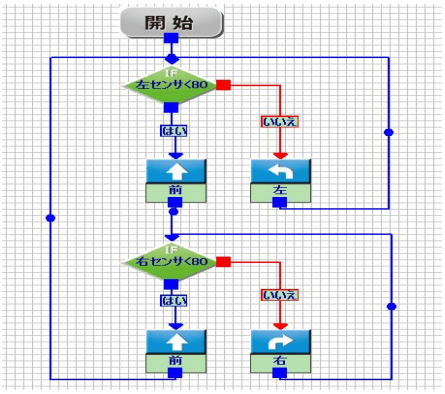

図 1　フローチャートによる記述のプログラム例

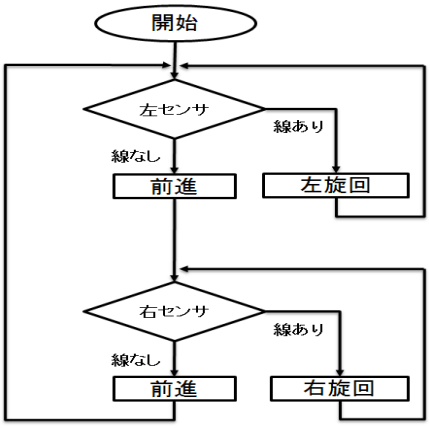

図 2　図 1 のプログラムのフローチャート

注 1)　http://www.vstone.co.jp/products/beauto_racer/

注 2)　http://www.elekit.co.jp/product/promo/kirobo/

2.2　タイルによる記述の例

プログラム例（IconWorks [注2)]）として図 3 に示したものは，2.1 と同じく，左右に取り付けられた 2 つの光センサ（センサ 1,センサ 2）を使ったライントレースプログラムである。このプログラムは，命令のタイルを線で結ぶことでプログラムの流れを表し，左上から右下へ流れるように書かれている。反復構造は，∞の絵が描かれた 2 種類のタイルで挟み込むことによって繰り返す範囲を指定している。分岐構造は，目の絵が描かれたタイルがセンサを表し，○×のタイルで判定後の分岐を示している。ライントレースでは，左右 2 つのセンサによる検知をそれぞれが繰り返しながら，プログラム全体を無限に繰り返す入れ子構造のプログラムとなる。これを，ループ端を用いたフローチャートで示すと図 4 のようになる。

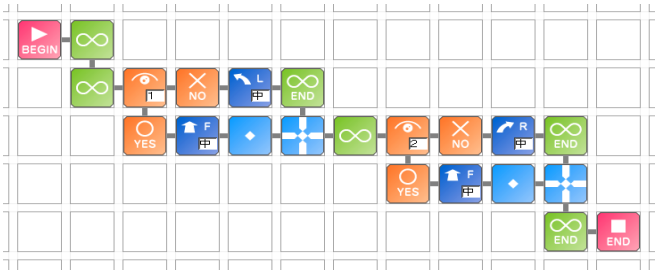

図 3　タイルによる記述のプログラム例

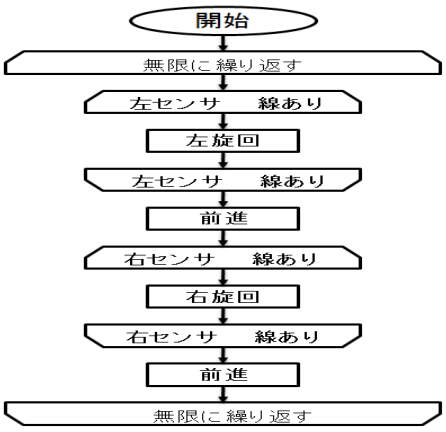

図 4　図 3 のプログラムのフローチャート

2.3　手続きの考え方に基づく記述の問題点

ライントレースのプログラムでは，前述の 2 種類のプログラムにあるように，センサの値による分岐構造が反復構造の中にあって繰り返されるという形を持つ。この形は，計測・制御プログラムで一般に見られる特徴でもある。

フローチャートによる記述は，矢印で表現され，全体の流れはよく分かるが，反復構造は読む側が線をたどってその意図を読み取らなくてはならない。プログラム初

心者である中学生が実際にプログラムを作ると，どこからどこまでを繰り返し，どこから次の段階の処理に進めるか，どのように矢印をつなげればよいかで悩むことが多い。とくに複数のセンサによる処理が入る場合は，どこに矢印を戻して反復すればよいかが分からなくなる場合がある。

タイルによる記述は，反復や分岐の構造が明示され，それぞれの部分的な構造は分かりやすくなっている。しかし，プログラム全体としては，図 3 のプログラムのように反復の中に反復が繰り返される入れ子構造になると，全体の構造を読み取るのが難しくなる。プログラム初心者は必要以上に複雑な入れ子構造を作る傾向があり，プログラム初心者である中学生の場合，複数の入れ子構造ができ，構造を理解できなくなる心配がある。

3. 状態遷移の考え方に基づくプログラムの可能性

手続きの考え方に基づく記述の問題を踏まえ，本研究では状態遷移の考え方を取り入れてプログラミングできないかを検討した。

3.1 状態遷移図

状態遷移とは，プログラムの状況が有限個の状態に分かれていて，条件に応じてその間を移り変わって行くという捉え方であり，その様子を図示したものが状態遷移図である。制御工学においては，状態遷移図を使って分かりやすく動作記述する手法が用いられている[8]。ライントレースプログラムを状態遷移図で示すと図 5 となる。この中で，角丸四角形が対象のシステムの「状態」を表している。矢印線によって，センサ値などの条件によって状態が変わることを表している。計測・制御プログラムの基本構造である反復は一つの状態に集約されている。この状態遷移図を使うことによって，プログラム初心者である中学生にとっても理解しやすいプログラムができると考えられる。

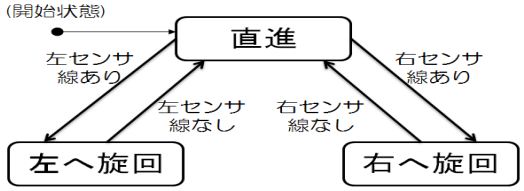

図 5 ライントレースプログラムの状態遷移図

3.2 状態遷移の考え方に基づいたプログラム

状態遷移図を用いたロボット制御の実践としては，千田らのロボットを用いた初心者のためのプログラミング教育[9]がある。千田らは，高専 5 年生に状態遷移図を用いた制御教育を実施した結果について，「状態遷移の考え方を取り入れることでロボットの制御構造が非常に見通し良くなり，容易にプログラムを作成することが可能となる」と報告している。

状態遷移の考え方に基づいたプログラミング言語の例としては，ロボット「梵天丸」のプログラミング言語「まきもの」[10]がある。「まきもの」は，ロボットの動きを状態遷移として理解できることを考慮して設計されている。「まきもの」の状態（だん＝段）にある動作の命令は「とまれ」「みぎまわれ」「ぜんしん」など 1 つだけ入れることができ，停止や前進などの動作状態を維持するようになっている。次の状態に移るための条件は，それぞれの状態（だん＝段）の最後に書かれている。

しかし「まきもの」は，条件によって動作を分岐したり，異なる動作に移る数だけ，新しい状態（段）を定義して呼び出す必要がある。そのため，課題が少し複雑になると，状態の数が増えてしまい，全体を理解しづらくなるという問題がある。図 6 は，後述する課題 2 のプログラム（図 14）を「まきもの」で記述したプログラムである。「前進状態」「後退状態」「回転状態」という，細かい動作ごとに状態を分けて記述する必要があるため，「壁に衝突したら反転して前進を続ける」という，行いたい動作の目的が読み取りにくくなっている。

そこで，状態遷移の考え方で，プログラム全体の流れが読み取りやすいプログラミング言語を検討した。

```
：前進のだん
    ぜんしん
    まえだ：後退のだん
：後退のだん
    こうしん
    １０：回転のだん
：回転のだん
    みぎまわれ
    １２：前進のだん
```

図 6 「まきもの」のプログラム例

4. プログラミング言語 Rtoys

状態遷移の考え方に基づいてロボット制御を記述するプログラミング言語 Rtoys[11]を，筆者の一人が開発している。Rtoys は学校現場で実践の多いビジュアルプログラミング環境のスクイーク Etoys[12]をロボット制御に対応させ，状態遷移の考え方でプログラミングできるようにしたものである。本論文では，この Rtoys を評価する。比較実験により評価するため，通常の手続きプログラミングの手法に基づいた Rtoys も試作した。以

下では，状態遷移の考え方に基づく方を「状態 Rtoys」，手続きの考え方に基づく方を「手続き Rtoys」と呼ぶ。

4.1 状態 Rtoys

状態 Rtoys によるライントレースプログラムを図 7 に示す。このプログラムは図 5 の状態遷移図をプログラムにしたものである。ロボットの動作状態を表すウィンドウが 3 つあり，それぞれ「直進」「左へ旋回」「右へ旋回」という状態名が付けられている。それぞれの「状態」の中には，「前進」「左回り」などのロボットの動作命令と，分岐命令を作るための「テスト」，光センサの値を指定するための「アナログセンサ」，状態を遷移させるための命令「状態を変える」が入っている。

状態 Rtoys では，ある一定の動作を繰り返す反復を 1 つの「状態」としている。「状態」の中には，「前進」「テスト」などの命令を複数入れることができ，それらは繰り返し実行される。複雑なプログラムは複数の状態に分割し，状態遷移によって結ぶ形で記述する。複雑なプログラムになっても，状態という単位で分割でき，全体的な流れが見やすく整理できるという特徴を持つ。

状態 Rtoys のプログラム画面を図 8 に示す。画面の右には使用できる命令が表示され，下にはロボットの動きがシミュレーション表示されている。

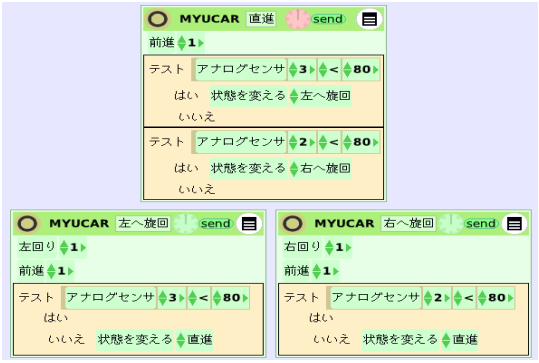

図 7 状態 Rtoys によるライントレースプログラム

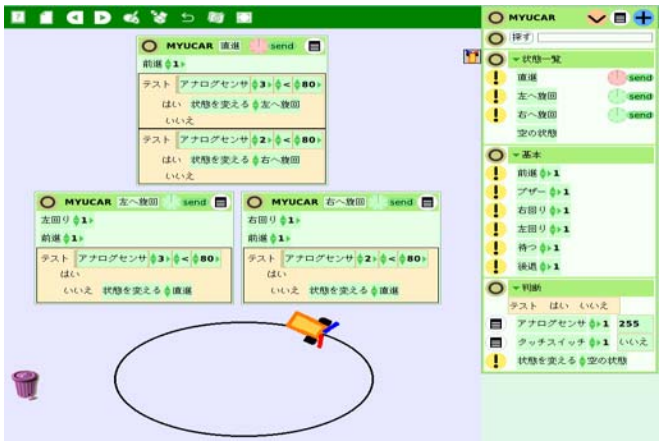

図 8 状態 Rtoys のプログラム画面

4.2 手続き Rtoys

手続き Rtoys のプログラム画面も，構成は図 8 と同じであるが，使用できる命令の一部が異なる仕様となっている。

光センサ（アナログセンサ）を使ったライントレースプログラムは，手続き Rtoys では図 9 のようになる。2.2 の図 4 のフローチャートに相当する考え方でプログラムを記述することができる。

図 9 手続き Rtoys によるライントレースプログラム

手続き Rtoys のプログラムは，2.3 の中でタイルによる記述の問題点としてあげたように，複数の入れ子ができ，全体の構造が読み取りにくくなる問題がある。この入れ子構造をなくす方法として，手続きの考え方に基づくプログラミングでは一般的に，break 文などのループから脱出する方法が用意されている。手続き Rtoys においても，ループから脱出する命令として「繰り返し脱出」を用意した。図 10 に示すプログラムは，触覚センサ（タッチセンサ）を使って，1 回目と 2 回目の接触で異なる動作をさせる場合である。前進している状態で，触覚センサに触れると後退に切り替わり，もう一度触覚センサに触れると前進に切り替わるプログラムを示している。「繰り返し脱出」を使わないプログラムは図 10 左のように入れ子構造になり，「繰り返し脱出」を使った入れ子構造でないプログラムは図 10 右のようになる。このように手続き Rtoys は，2 つの方法でプログラムでき，課題によって生徒が考えやすい方で作ることができる。

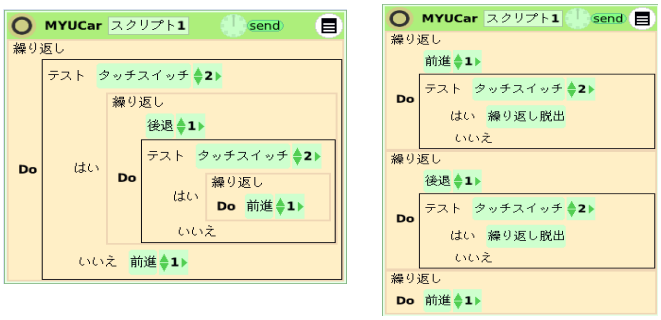

図 10　入れ子構造の場合(左)と繰り返し脱出の場合(右)の違い

5. 検証授業

技術・家庭科の計測・制御学習の授業で，実際に状態 Rtoys と手続き Rtoys を使い，学習効果を検証した。

5.1　検証授業の仮説

状態 Rtoys と手続き Rtoys を使った制御プログラムにおいて，次の仮説の検証を試みた。

仮説: 状態遷移の考え方に基づく方法を取り入れることにより，生徒は自律型ロボットが複数の状態で動作するような実用的なプログラムが作れるようになる。

5.2　検証に使用した自律型ロボットとソフトウェア

今回の検証に用いた自律型ロボットは，左右に触覚センサ(タッチセンサ)のついた車型の自律型ロボット(図 11）である。検証に用いたソフトウェアは，本来の状態遷移の考え方に基づく状態 Rtoys と，今回の比較実験のために用意した手続き Rtoys である。

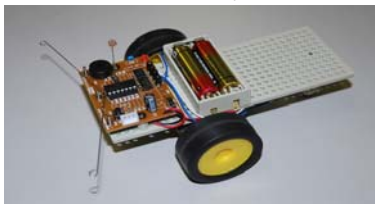

図 11　検証に使用した車型の自律型ロボット

5.3　授業概要・計画

検証授業は，筆者の一人が勤務する公立中学校の 3 年生 1 クラスを 2 つに分けて実施している技術・家庭科で行った。それぞれのクラスに同じロボットと課題を与え，ソフトウェアだけを変えた比較実験とした。データは 2 回以上欠席した生徒を除く，各クラス 16 名ずつを分析した。授業の概要を表 1 に示す。

第 1 時は，Etoys のプログラミング方法を習得させるための内容で，ロボットを使わずに実施した。第 2 時以降はそれぞれのクラスで状態 Rtoys と手続き Rtoys を使用し，実際にロボットにプログラムを転送して動作させながら授業を進めた。以下では，状態 Rtoys を使って授業をしたクラスを「状態クラス」，手続き Rtoys を使って授業をしたクラスを「手続きクラス」と呼ぶ。表 2 に授業計画を示す。

課題は，ロボットの特徴である触覚センサを使った迷路脱出の授業を想定したものとし，学習のステップや難易度を考慮して設定した。授業方法は，考える時間を十分に与え，できる限り生徒の力だけで解決させるようにした。前時の理解が必要な授業では，最初に前時の課題の解説をした後，新しい課題に取り組ませた。

表 1　検証授業概要

教科：技術・家庭科必修授業　　学年：中学 3 年生
時間数：6 単位時間　(1 単位時間は 50 分)

	状態クラス	手続きクラス
使用ソフトウェア	状態 Rtoys	手続き Rtoys
使用ロボット	自律型ロボット	自律型ロボット
受講者数	16 名	16 名

表 2　授業計画

第 1 時	Etoys によるアニメーション
第 2 時	課題 1：ロボットが円を描くプログラム
第 3 時	課題 2：壁に衝突すると反対に向きを変えるプログラム
第 4 時	課題 3：壁に衝突すると後退するプログラム
第 5 時	課題 4,5：触覚に触れるたびに動きを変えるプログラム
第 6 時	課題 6：(自由制作) 触覚に触れるたびに動きを変える作品

5.4　授業内容

授業内容を指導者の授業記録をもとに説明する。

第 1 時：Etoys によるアニメーション

第 1 時は，Etoys の絵を操作する機能の使い方や基本的な命令の作り方を指導した。自分の描いた図形に命令を入れ，動きや音を楽しむアニメーションを作らせた。

第 2 時：円を描く課題

第 2 時は，ロボットへのプログラムの転送方法，ロボットの動かし方を中心に指導した。また，それぞれの Rtoys にある命令やシミュレーションについて簡単に説明した。課題は，「課題 1：ロボットが円を描くプロ

グラム」とし，できた生徒には正方形や三角形を描くプログラムにも挑戦させた。ロボットが同じ動作（円や正方形を描く）を繰り返すプログラムとし，1つの繰り返しの中で考えられる課題にした。課題 1 の正解例を図 12 に示す。

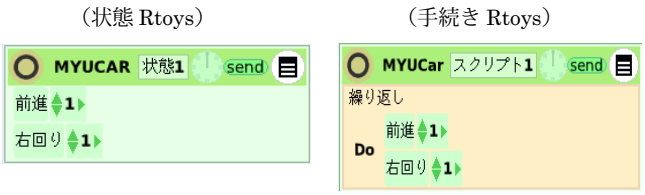

図 12　課題 1 の正解プログラム例

第 3 時：壁に衝突すると反対に向きを変える課題

第 3 時は，触覚センサの働きと分岐命令の使い方を指導した後，「課題 2：壁に衝突すると反対に向きを変えるプログラム」（図 13）を考えさせた。ロボットがある条件のときに違う動作に変わるプログラムとし，1つの「状態」の中で考えることができる課題として設定した。正解例を図 14 に示す。

課題 2：
前進しながら，何かに衝突すると反対向きに向きを変えるプログラムを作り，ロボットを動かしてみましょう。

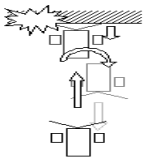

図 13　課題 2：壁に衝突すると反対に向きを変える課題

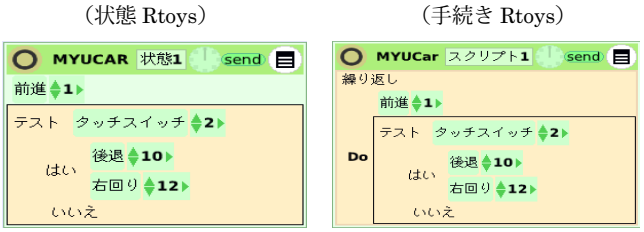

図 14　課題 2 の正解プログラム例

第 4 時：壁に衝突すると後退する課題

第 4 時は，分岐命令の使い方を復習した後，「課題 3：壁に衝突すると後退するプログラム」（図 15）を考えさせた。ロボットがある動作を繰り返しながら，ある条件で別の動作の繰り返しに移るプログラムとし，2 つの「状態」の中で考えることを想定して設定した。授業で作成した課題通りに動く生徒プログラムを図 16 に示す。

課題 3：
前進しながら，何かに衝突すると後退し続けるプログラムを作り，ロボットを動かしてみましょう。

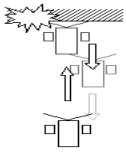

図 15　課題 3：壁に衝突すると後退する課題

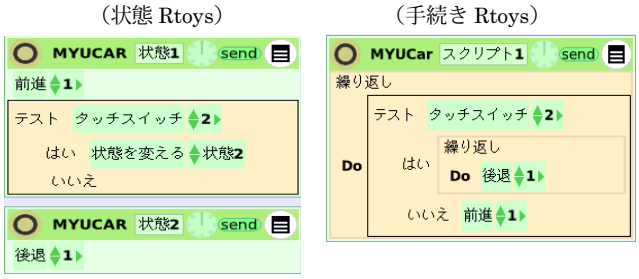

図 16　課題 3 において生徒が作成したプログラム例（正解）

第 5 時：触覚に触れるたびに動きを変える課題

第 5 時では最初に，課題 3 での「状態」の考え方を，状態 Rtoys は「状態を変える」，手続き Rtoys は「繰り返し脱出」を使って指導した。次に，「課題 4,5：触覚に触れるたびに動きを変えるプログラム」（図 17）を考えさせた。ロボットが 2 つ又は 3 つの動作をするプログラムとし，複数の「状態」を考える課題として設定した。

課題 4 の生徒プログラムを図 18 と図 19 に示す。状態 Rtoys（図 18）では右回りと静止の状態の 2 種類の状態を作成し，右回りの状態で左の触覚センサが接触すると静止の状態に切り替えるように考えて解決している。手続き Rtoys（図 19 左）では，上から手順を考え，右回りを繰り返している状態で，左の触覚センサが接触すると「繰り返し脱出」命令で，繰り返しから抜け出し，静止の状態に切り替えるように考えて解決している。

課題 4：次のストーリーでプログラムを作って，ロボットを動かしてみましょう。
1）初めロボットは止まっています。
2）右側の触覚が触られると右側に回り始めます。
3）2）で回っている時，左側の触覚を触れると止まります。

課題 5：次のストーリーでプログラムを作って，ロボットを動かしてみましょう。
1）初めロボットは止まっています。
2）右側の触覚が触られると右側に回り始めます。左側の触覚が触られると左側に回り始めます。
3）2）で回っている時，反対側の触覚を触れると止まります。

図 17　課題 4，5：触覚に触れるたびに動きを変える課題

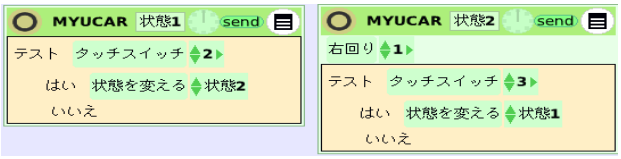

図 18　課題 4 において生徒が作成したプログラム例（正解）（左右でひとつのプログラム）（状態 Rtoys）

(繰り返し脱出の場合)　(入れ子構造の場合)

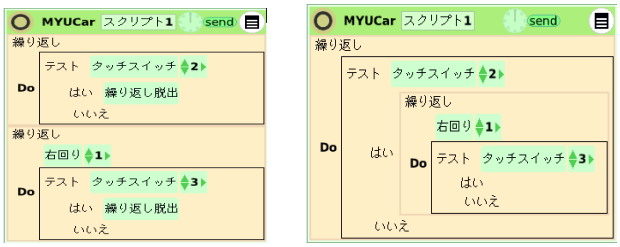

図 19　課題 4 において生徒が作成したプログラム例(正解)
(手続き Rtoys)

第 6 時：触覚に触れるたびに動きを変える作品

最後の授業は，今までの学習内容を使った自由制作の時間とした。課題は「課題 6：触覚に触れるたびに動きを変える作品」とし，数多く動きが切り替わる作品にするように指導した。

6. 検証授業の結果

6.1　課題の達成率

それぞれの課題の達成率を表 3 に示す。達成人数は，生徒が授業者の前でプログラムを実行し，授業者が課題通りに動作することを確認した人数を示している。それぞれのクラスの達成率を比較すると，課題 3 を除き，状態クラスの方が，達成率が高い結果となっている。課題 1 と複雑な動きをする課題 4,5 に対しては，有意差が見られた（1%水準で有意）。

表 3　課題の達成率

	状態クラス	手続きクラス	p 値	有意差
課題 1	94%（15/16）	50%（ 8/16）	0.0077	p<0.01
課題 2	94%（15/16）	73%（11/15）	0.15	無
課題 3	13%（ 2/16）	31%（ 5/16）	0.20	無
課題 4	75%（12/16）	19%（ 3/16）	0.0019	p<0.01
課題 5	44%（ 7/16）	0.0%（ 0/16）	0.0034	p<0.01

・達成率の（ ）内は（達成人数 / 出席人数），検定方法はフィッシャーの正確確率検定

6.2　状態の理解についての観察

「状態」の考えを初めて必要とする「課題 3：壁に衝突すると後退するプログラム」の授業では，生徒の問題解決の過程を調査した。この課題は，前進と後退でロボットの状態が異なることを考える必要がある。しかし，1 つの状態だけで考えると，前進する中で，触覚センサが触れているときのみ後退し，触れていないときは前進するプログラムになる。触覚センサが離れた瞬間に前進に切り替わり，ロボットは壁の前で前後の動きを繰り返す動作となる。生徒のプログラムを調査したところ，両方のクラスで全ての生徒が，最初はこのような前後の動きを繰り返すプログラムを作っていた。実際に生徒が最初に作ったプログラムを図 20 に示す。

時間を十分与えても，解決できない生徒が多いため，どの段階まで考えているかを調べた。生徒からは，「後退する時間が終了すると，プログラムの最初の前進に戻り，前進してしまう」「スイッチが入っている間だけしか，後退命令が効かない」という意見があった。挙手により確認したところ，ほぼ全員の生徒が，「原因は分かっているが，解決するためのプログラムが作れない」状態であった。そこで次時（第 5 時）の授業の最初に，「状態」の考え方をそれぞれのクラスで指導した。課題 3 のプログラムを例に，状態 Rtoys は「状態を変える」を使って 2 つの状態に分けることを，手続き Rtoys は「繰り返し脱出」を使って次の状態に移ることを指導した。

(状態 Rtoys)　(手続き Rtoys)

図 20　課題 3 において生徒が最初に作成したプログラム例
(不正解)

6.3　生徒作品の分析

第 6 時に制作した課題 6 の生徒作品を分析した。課題は「触覚に触れるたびに動きを変える作品」である。触覚センサで「前進」「右回り」などの動作が切り替わる回数（動作数）を調べたところ，図 21 のようになった。手続きクラスにおける動作数の分布は左右対称でなく，外れ値も存在することから，正規分布を仮定した t 検定ではなく，Wilcoxon の順位和検定を行ったところ p=0.024 で，5%水準で 2 つの群に有意な差が認められた。

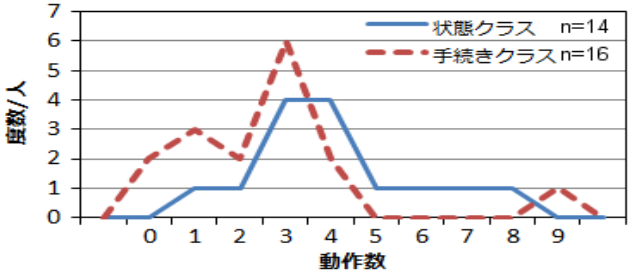

図 21　触覚センサによる動作数の人数分布

同じ動作数の生徒作品を図 22 と図 23 に示す。状態 Rtoys の作品（図 22）は，状態ごとに分けられて動作が分かりやすくなっているが，手続き Rtoys の作品（図 23）は，入れ子構造になっている部分では動作の内容が読み取りにくくなっている。また，プログラムを見て分かるように，同じような動作をさせる場合でも，考える複雑さは大きく異なると思われる。

図 22　状態 Rtoys の生徒作品　（動作数 4）

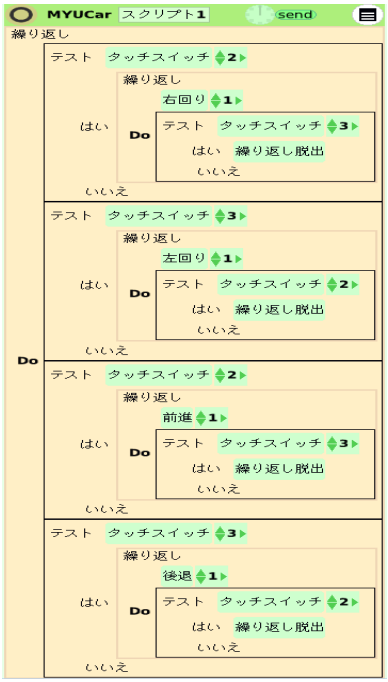

図 23　手続き Rtoys の生徒作品　（動作数 4）

7. 考察

課題の達成率（表 3）から，状態クラスは，初めて複数の状態を扱う課題 3 はできていないが，いちど理解した後は，続く課題を高い割合で完成している。逆に，手続きクラスは課題が進むに連れて完成できない生徒が増えている。状態 Rtoys は，ロボットが複数の状態を必要とする複雑な動きをする課題に対して有効であることが分かる。

課題 3 について，状態クラスの正解者が少ないのは複数の状態を考える初めての課題のため，別の状態が必要になるという考え方ができなかったためであると考えられる。一方，手続きクラスの生徒は，図 16 の生徒プログラムでは，反復から脱出して別の状態を考えているのではなく，1 つの反復の中に更に反復を入れ子構造にすることで解決している。他の生徒も同じで，「繰り返し脱出」を使って別の状態にする考え方の解答はなかった。また，最初は全員が状態を考えないプログラムを作っていたことや，後に原因は分かってきたが解決できなかったことから，「状態」のとらえ方や「状態が遷移する」「反復から脱出して別の状態にする」という考え方は，課題を解いている中で自然と思いつくものではなく，指導する必要のある内容であることが分かった。今回の授業では，実際にプログラムを作る過程で不具合を体験した後に，状態の考え方を指導した。

状態の考え方を指導した後の生徒作品の分析では，状態 Rtoys の方はロボットのそれぞれの状態に分け，自分の考えを整理しながら作品を作っている。一方，手続き Rtoys の方は繰り返し脱出を使った作品でも，動作を整理できていない作品も数多く見られた。このことから，状態 Rtoys の方が，状態の考え方が身につきやすく，自分の考えを整理しやすい言語であると考えられる。そして，その結果として，状態 Rtoys の方が触覚センサに，数多く動きが切り替わる制御プログラムを作ることができていると考えられる。

8. まとめ

市販されている計測・制御学習用の教材を中心に，手続きの考え方に基づくプログラムの特徴と問題点について検討し，問題点を克服する試みとして状態遷移の考えを取り入れたプログラミング方法を提案した。検証では，中学生を対象に，自律型ロボットを用いた計測・制御のプログラミング学習を実施した。従来型の言語を用いた対照群と比較した結果，次のことが分かった。

(1) 手続き型のプログラミングに比べ，状態遷移の考え方に基づく方法は，複雑な動きをするプログラムを記述できるようになる。

(2) ただし，状態遷移の考え方は自然に気付くものではなく，指導する必要がある。

今後の検証については，2つの方向で発展させたいと考えている。提案した状態 Rtoys については，今回扱ったライントレースや触覚センサを使った壁衝突に加え，3軸制御ロボットなど機能の異なる他のロボット教材でも実験してみたい。また，比較の対象として，今回使った手続き Rtoys だけでなく，市販教材に付属するソフトウェアによるプログラミングとも比較を行いたい。

学習の効果については，状態遷移の考え方を導入することで，より複雑な動きを行わせる課題を扱えるようになることを確認した。複雑な動きをする課題は，複数の状態を持って動作していることから，複数の状態を持つ情報機器やシステムの仕組みを理解する学習に発展できる可能性が考えられる。今後は，中学生が課題においてどの程度まで複雑な動きを行わせられるか，その学習を通してどのような技術を理解できるようになるかを調べていきたい。

参考文献

1) 文部科学省:中学校学習指導要領 第2章 各教科第8節技術・家庭 (2008)
2) 紅林秀治・江口啓・兼宗進：制御プログラム学習における中学生の学習効果，日本産業技術教育学会誌，第51巻，第4号，pp.301-309 (2009)
3) 古平真一郎・坂本弘志・針谷安男：自律型ロボット教材を用いた「プログラムによる計測・制御」学習の授業実践に基づく学習効果の検証，日本産業技術教育学会誌，第51巻，第4号，pp.285-292 (2009)
4) 伊藤陽介・森誉範・菊地章 他：「プログラムと計測・制御」のためのロボット学習材の開発と実践，日本産業技術教育学会誌，第 49 巻，第 3 号，pp.213-221 (2007)
5) 嶋田彰子・山菅和良・針谷安男 他：自律型ロボット教材を活用したプログラムと計測・制御学習に関する授業方法の開発と評価，日本産業技術教育学会誌，第49巻，第4号，pp.297-305 (2007)
6) 伊藤陽介・石塚仁志・大泉計 他：ロボカップジュニア・レスキューを題材とする情報技術学習の提案，日本産業技術教育学会誌，第50巻，第2号，pp.59-67 (2008)
7) 井戸坂幸男・久野靖・兼宗進：自律型ロボット教材の評価と授業，日本産業技術教育学会誌，第53巻，第1号，pp.9-16 (2011)
8) 滝田啓司・酒井充・米田政明 他：オブジェクト指向と状態遷移モデルによるシーケンス制御用言語，情報処理学会研究報告，91-SE-83，pp.139-146 (1992)
9) 千田和範・野口孝文・梶原秀一 他：ロボットを用いた初心者のためのプログラミング教育，電子情報通信学会技術研究報告. ET，教育工学 106(166)，pp.21-24 (2006)
10) 岩本正敏・水谷好成：ロボット教材を活用した情報基礎教育，電子情報通信学会技術研究報告. ET，教育工学 101(309)，pp.31-38 (2001)
11) Hiroyuki Aoki, DongHee Park, WonGyu Lee：Robot Programming with Squeak “Rtoys” - Connection between Drawing Worlds and the Real World, International Conference on Technology Education in the Asia Pacific Region Conference Proceedings, pp.390-398 (2009)
12) Squeak Etoys：http://www.squeakland.org/

Abstract

The topic of "measurement and control" in technology education of lower high schools is often treated with programming activity for autonomous mobile robots. Commercially available teaching materials for autonomous mobile robots are accompanied by visual programming environments, based either on flowchart style or tile-scripting style both based on procedural programming model, without explicit notation of program states. However, in programming for measurement and control, explicit notion of "states" are often helpful to craft correct program. We have developed "Rtoys", which is reformulation of "Squeak Etoys" with addition of state transition. Our classroom evaluation indicates that "Rtoys" with state transition model is effective for building correct sensor-equipped robot control programs.

Key words：Measurement and control, Autonomous robot, Squeak Etoys, Program